

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	882	717/124.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 16:23
S2	291	717/124.ccls. and (tree or hierarch\$6)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 16:23
S3	5	717/124.ccls. and (collection near2 test\$3) and correctness	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 16:24
S4	72	717/124.ccls. and correctness	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 16:25
S5	30	717/124.ccls. and (parameter same fail\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 16:30
S6	25	717/124.ccls. and ((argument or parameter) same fail\$3) and (hierarch\$7 or tree or "control flow" or cfg or path or (control\$4 near3 direct\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:06
S7	9	((fault or fail\$3) near3 (event adj handl\$3)) and ((argument or parameter) same fail\$3) and (hierarch\$7 or tree or "control flow" or cfg or path or (control\$4 near3 direct\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:10
S8	0	test\$3 near5 ((fault or fail\$3) near3 (event adj handl\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:11
S9	0	(717/12?.ccls. or 717/13?.ccls. or 714/3?.ccls. or 714/51.ccls.) and test3 and (((fault or fail\$3) near3 (event adj handl\$3)) same (parameter or value or argument)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:13

EAST Search History

S10	0	(717/12?.ccls. or 717/13?.ccls. or 714/3?.ccls. or 714/51.ccls.) and test3 and (event adj handl\$3) same (parameter or value or argument)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:13
S11	55	(717/12?.ccls. or 717/13?.ccls. or 714/3?.ccls. or 714/51.ccls.) and (event adj handl\$3) same (parameter or value or argument)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:19
S12	3	test adj assertion and test adj execution and event	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/16 17:20


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#)
[Search:](#) [The ACM Digital Library](#) [The Guide](#)
[+data +flow +testing](#)
[The ACM Digital Library](#)
[Feedback](#) [Report a problem](#) [Satisfaction](#)

Published since January 1990 and Published before
November 2003

Found **12,204**

Terms used **data flow testing**

Sort results by

[Save results to a Binder](#)

[Try an Advanced Search](#)

Display results

[Search Tips](#)

Try this search in [The ACM](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

1 [A demand-driven analyzer for data flow testing at the integration level](#)

Evelyn Duesterwald, Rajiv Gupta, Mary Lou Soffa

May 1996 **Proceedings of the 18th international conference on Software engineering; ICSE '96**

Publisher: IEEE Computer Society

Full text available: [pdf\(1.12](#)

[MB\)](#)

[Publisher](#)

[Site](#)

Additional Information: [full citation](#), [abstract](#), [reference](#)
[citations](#), [index terms](#)

Data-flow testing relies on static analysis for computing the definition-use pairs that satisfy the test case requirements for a program. When testing large programs, the individual procedures are first tested in isolation during unit testing. Integration testing is performed specifically to test the procedure interfaces. The procedures in a program are integrated and tested in several steps. Since each integration step requires data-flow analysis to determine the new test requirements, the accuracy of the test case requirements is improved.


Keywords: data flow analysis, data flow testing, definition-use pairs, demand-driven analyzer, exhaustive analyzer, incremental analyzer, incremental data-flow updates, integration testing, large program testing, overhead, performance, program procedure interfaces, program testing, static analysis, test case requirements, unit testing


2 Technical papers: testing II: Data flow testing as model checking

Hyoung Seok Hong, Sung Deok Cha, Insup Lee, Oleg Sokolsky, Hasan Ural

May 2003 **Proceedings of the 25th International Conference on Software Engineering**
ICSE '03

Publisher: IEEE Computer Society

Full text available:  pdf(1.00

MB) 


Publisher

Site

Additional Information: full citation, abstract, reference
index terms


This paper presents a model checking-based approach to data flow testing. We characterize data flow oriented coverage criteria in temporal logic such that the problem of test generation is reduced to the problem of finding witnesses for a set of temporal logic formulas. The capability of model checkers to construct witnesses and counterexamples allows test generation to be fully automatic. We discuss complexity issues in minimal test generation and describe heuristic test generation algorithm ...

3 The path-wise approach to data flow testing with pointer variables

 Delia I. S. Marx, Phyllis G. Frankl

May 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1996 ACM SIGSOFT international symposium on Software testing and analysis ISS '96**, Volume 21 Issue 3

Publisher: ACM Press

Full text available:  pdf(941.63 KB) Additional Information: full citation, abstract, reference
index terms


This paper describes a new approach to performing data flow testing on programs that use pointer variables and a tool based on this approach. Our technique is based on the observation that, under certain reasonable assumptions, we can determine which dereferenced pointers are aliased whenever control reaches a given program point *via particular path*. Furthermore, we can group together paths which behave similarly and represent them by regular expressions. The resulting test requirements ...

4 The effects of optimizing transformations on data-flow adequate test sets


 Mary Jean Harrold

October 1991 **Proceedings of the symposium on Testing, analysis, and verification T**

Publisher: ACM Press


Full text available:  [pdf\(858.51 KB\)](#) Additional Information: [full citation](#), [references](#), [index](#)

5 Data flow-based test adequacy analysis for languages with pointers

 Thomas J. Ostrand, Elaine J. Weyuker

October 1991 **Proceedings of the symposium on Testing, analysis, and verification T**

Publisher: ACM Press



Full text available:  [pdf\(1.19 MB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)

6 An exact array reference analysis for data flow testing

István Forgács

May 1996 **Proceedings of the 18th international conference on Software engineering; ICSE '96**

Publisher: IEEE Computer Society

Full text available:  [pdf\(1.05 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [reference citings](#), [index terms](#)
[Publisher Site](#)

Data-flow testing is a well-known technique, and it has proved to be better than the commercially-used branch testing. The problem with data-flow testing is that, apart from scalar variables, only approximate information is available. This paper presents an algorithm that precisely determines the definition-use pairs for arrays within a large domain. The numerous methods addressing the array data-flow problem; however, these methods are used in the optimization or parallelization of p ...

Keywords: approximate information, arrays, data flow analysis, data flow testing, definition-use pairs, exact array reference analysis, formulae negation avoidance, precision method, program optimization, program parallelization, program path execution, program testing


7

An analytical comparison of the fault-detecting ability of data flow testing techniques

Phyllis G. Frankl, Elaine J. Weyuker

May 1993 **Proceedings of the 15th international conference on Software Engineering
ICSE '93**

Publisher: IEEE Computer Society Press

Full text available:  pdf(1.05
MB)


Additional Information: [full citation](#), [references](#)

8 A static measure of a subset of intra-procedural data flow testing coverage based on node coverage

Ettore M. Merlo, Giuliano Antoniol

November 1999 **Proceedings of the 1999 conference of the Centre for Advanced Studies
on Collaborative research CASCON '99**

Publisher: IBM Press

Full text available:  pdf(225.32 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


In the past years, a number of research works, which have been mostly based on pre a post dominator analysis, have been presented about finding subsets of nodes and edge (called "unrestricted subsets") such that their traversal during execution (if feasible) exercises respectively all feasible nodes and edges in a Control Flow Graph (CFG). The paper presents an approach to statically measure a subset of intra-procedural data flow uses") coverage obtained by exercising an "unrestricted s ...

9 Performing data flow testing on classes

 Mary Jean Harrold, Gregg Rothermel

December 1994 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2nd
ACM SIGSOFT symposium on Foundations of software engineering
SIGSOFT '94, Volume 19 Issue 5**

Publisher: ACM Press

Full text available:  pdf(1.01 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The basic unit of testing in an object-oriented program is a class. Although there has been much recent research on testing of classes, most of this work has focused on black-box approaches. However, since black-box testing techniques may not provide sufficient coverage, they should be augmented with code-based or white-box techniques. Dataflow testing is a code-based testing technique that uses the dataflow relations in a program


guide the selection of tests. Existing dataflow testing t ...

10 Testing Java programs using dynamic data flow analysis

◆ A. S. Boujarwah, K. Saleh, J. Al-Dallal

March 2000 **Proceedings of the 2000 ACM symposium on Applied computing - Vol 1**
SAC '00

Publisher: ACM Press

Full text available:  [pdf\(226.61 KB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)


Keywords: Java, dynamic data flow analysis, instrumentation, object oriented programming, software testing

11 Augmenting data flow criteria for class testing

Pei Hsia, Xiaolin Li, David C. Kung

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Study on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available:  [pdf\(219.68 KB\)](#) Additional Information: [full citation](#), [abstract](#), [reference index terms](#)

A class is widely considered the basic unit of testing in object-oriented software. Although there has been much recent research on class testing, little attention has been paid to code-based class testing criteria. In this paper, we extend the traditional data flow testing techniques and propose three new code-based class testing criteria. These new criteria overcome the problems associated with existing data flow techniques. We also show that each of the new criteria is stricter than its ...




12 Modeling software for accurate data flow representation

Hasan Ural, Bo Yang

May 1993 **Proceedings of the 15th international conference on Software Engineering**
ICSE '93

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(941.86 KB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)

13 Data flow coverage and the C language J. R. Horgan, S. LondonOctober 1991 **Proceedings of the symposium on Testing, analysis, and verification T****Publisher:** ACM PressFull text available:  [pdf\(1.12 MB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)**14 The chaining approach for software test data generation** Roger Ferguson, Bogdan KorelJanuary 1996 **ACM Transactions on Software Engineering and Methodology (TOSI)**
Volume 5 Issue 1**Publisher:** ACM PressFull text available:  [pdf\(1.53 MB\)](#) Additional Information: [full citation](#), [abstract](#), [reference citations](#), [index terms](#), [review](#)

Software testing is very labor intensive and expensive and accounts for a significant portion of software system development cost. If the testing process could be automated, the cost of developing software could be significantly reduced. Test data generation in program testing is the process of identifying a set of test data that satisfies a selected testing criterion, such as statement coverage and branch coverage. In this article we present a chaining approach to automate test data generation.

Keywords: data dependency, dynamic analysis, heuristics, program execution**15 Session 8B: embedded systems power management and validation: A data flow fault coverage metric for validation of behavioral HDL descriptions**

Qiushuang Zhang, Ian G. Harris

November 2000 **Proceedings of the 2000 IEEE/ACM international conference on Computer-aided design ICCAD '00****Publisher:** IEEE PressFull text available:  [pdf\(65.11 KB\)](#) Additional Information: [full citation](#), [abstract](#), [reference citations](#)

Behavioral HDL descriptions are commonly used to capture the high-level functionality of a system.

hardware circuit for simulation and synthesis. The manual process of creating a behavioral description is error prone, so significant effort must be made to verify the correctness of behavioral descriptions. Simulation-based validation and formal verification are both techniques used to verify correctness. We investigate validation because formal verification techniques are frequently intractable for large ...

16 Automated test data generation for programs with procedures

◆ Bogdan Korel

May 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1996 ACM SIGSOFT international symposium on Software testing and analysis ISS '96**, Volume 21 Issue 3

Publisher: ACM Press

Full text available:  pdf(716.54 KB) Additional Information: [full citation](#), [abstract](#), [reference citations](#), [index terms](#)


Test data generation in program testing is the process of identifying a set of test data that satisfies a selected testing criterion, such as, statement coverage or branch coverage. The existing methods of test data generation are limited to unit testing and may not efficiently generate test data for programs with procedures. In this paper we present an approach to automated test data generation for programs with procedures. This approach builds on current theory of execution-oriented test ...

17 Evaluation of predicated array data-flow analysis for automatic parallelization

◆ Sungdo Moon, Mary W. Hall

May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '99**, Volume 34 Issue 8


Publisher: ACM Press

Full text available:  pdf(1.54 MB) Additional Information: [full citation](#), [abstract](#), [reference citations](#), [index terms](#)



This paper presents an evaluation of a new analysis for parallelizing compilers called *predicated array data-flow analysis*. This analysis extends array data-flow analysis for parallelization and privatization to associate predicates with data-flow values. These predicates can be used to derive conditions under which dependences can be eliminated and privatization is possible. These conditions can be used both to enhance compile-time analysis and to introduce run-time tests that guard safe ...

18 Predicated array data-flow analysis for run-time parallelization

◆ Sungdo Moon, Mary W. Hall, Brian R. Murphy

July 1998 **Proceedings of the 12th international conference on Supercomputing ICS****Publisher:** ACM PressFull text available:  [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)**19** A design for testability technique for RTL circuits using control/data flow extraction


Indradeep Ghosh, Anand Raghunathan, Niraj K. Jha

January 1997 **Proceedings of the 1996 IEEE/ACM international conference on Computer aided design ICCAD '96****Publisher:** IEEE Computer SocietyFull text available:  [pdf\(172.02 KB\)](#)  Additional Information: [full citation](#), [abstract](#), [reference citings](#), [index terms](#)
[Publisher Site](#)

In this paper, we present a technique for extracting functional (control/data flow) information from register transfer level (RTL) controller/data path circuits and illustrate its use in design for hierarchical testability of these circuits. This testing procedure and design for testability (DFT) technique is general enough to handle RTL control flow intensive circuits like protocol handlers as well as data flow intensive circuits like digital filters. This makes the combined controller-data path ...





20 Refining data flow information using infeasible paths

◆ Rastislav Bodík, Rajiv Gupta, Mary Lou Soffa

November 1997 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th European conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering FSE '97/FSE-5, Volume 22 Issue 6****Publisher:** Springer-Verlag New York, Inc., ACM PressFull text available:  [pdf\(1.48 MB\)](#) Additional Information: [full citation](#), [references](#), [citing index terms](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Player](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service\)](#)
[Search:](#) [The ACM Digital Library](#) [The ACM Digital Library](#)
[+test +hierarchy +expected](#)

[Feedback](#) [Report a problem](#)

Published since January 1990 and Published before
November 2003

Terms used **test hierarchy expected**

Sort results
by

relevance

[Save results to a Binder](#)

Try an [Advanced Search](#)

[Search Tips](#)

Try this search

Display
results

expanded form

☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Results

1 [Experiences in automating the testing of SS7 signalling transfer points](#)

[PDF](#) Tim Moors, Malathi Veeraraghavan, Zhifeng Tao, Xuan Zheng, Ramesh B
July 2002 **ACM SIGSOFT Software Engineering Notes**, **Proceedings of
SIGSOFT international symposium on Software testing and
'02**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: [PDF \(271.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)


Signalling System 7 (SS7) is widely used for telephone signalling. Service providers frequently test their Signalling Transfer Points (STPs), which switch SS7 protocol conformance and interoperability. This paper describes a system that analyzes the data collected during STP tests. It consists of files that describe the expected behavior during the test, and Perl code that translates this into a program that can search the ...

Keywords: SS7, STP, automation, signaling system 7, signalling system

2 [Session 3: CLODs: dual hierarchies for multiresolution collision detection](#) Miguel A. Otaduy, Ming C. Lin

June 2003 **Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Computer Graphics and Geometry processing SGP '03**

Publisher: Eurographics Association

Full text available:  [pdf\(1.19 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)


We present "*contact levels of detail*" (CLOD), a novel concept for multi-resolution detection. Given a polyhedral model, our algorithm automatically builds both a multiresolution representation of the original model and its bounding hierarchy for accelerating collision queries. We have proposed various error models including object-space errors, velocity dependent gap, screen-space error combinations. At runtime, our algorithm uses these error models ...

3 Automatic labeling of semantic roles

Daniel Gildea, Daniel Jurafsky

September 2002 **Computational Linguistics**, Volume 28 Issue 3

Publisher: MIT Press

Full text available:  [pdf\(573.51 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)


We present a system for identifying the semantic relationships, or semantic constituents of a sentence within a semantic frame. Given an input sentence, word and frame, the system labels constituents with either abstract semantic roles AGENT or PATIENT, or more domain-specific semantic roles, such as MESSAGE, and TOPIC. The system is based on statistical classifiers trained on 50,000 sentences that were hand-labeled.

4 Applying predication to efficiently handle runtime class testing

 Chris Sadler, Sandeep K. S. Gupta, Rohit Bhatia

March 2000 **ACM SIGARCH Computer Architecture News**, Volume 2

Publisher: ACM Press

Full text available:  [pdf\(741.25 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

Runtime class testing is a technique whereby virtual function calls are translated into statically-bound function calls through a series of conditional branches. Through this transformation, the overhead of virtual function calls can be significantly reduced. The drawback of these tests is that by relying on conditional branches, the instruction-level parallelism (ILP) is limited and the mispredict penalties


high. We show that by using predication during cla ...

5 Cheops: a compact explorer for complex hierarchies

Luc Beaudoin, Marc-Antoine Parent, Louis C. Vroomen

October 1996 **Proceedings of the 7th conference on Visualization '96 VI**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(1.14 MB\)](#) Additional Information: [full citation](#), [reference](#), [index terms](#)


Keywords: focus+context techniques, graphical browser, hierarchical re information visualization and exploration

6 On testing hierarchies for protocols

Deepinder P. Sidhu, Howard Motteler, Raghu Vallurupalli

October 1993 **IEEE/ACM Transactions on Networking (TON)**, Volume 1

Publisher: IEEE Press


Full text available:  [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [reference](#), [index terms](#)

7 Efficient type inclusion tests

 Jan Vitek, R. Nigel Horspool, Andreas Krall

October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM conference on Object-oriented programming, systems, languages and applications OOPSLA '97**, Volume 32 Issue 10

Publisher: ACM Press


Full text available:  [pdf\(2.39 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

A type inclusion test determines whether one type is a subtype of another. Testing techniques exist for single subtyping, but not for languages with multiple inheritance. To date, the fast constant-time technique relies on a binary matrix encoding the subtype relation with quadratic space requirements. In this paper, we present three encodings: the subtype relation, the *packed encoding*, the *bit-packed encoding* and the *trie encoding*. These encodings have ...

8 Efficient maintenance and self-collision testing for Kinematic Chains

◆ Itay Lotan, Fabian Schwarzer, Dan Halperin, Jean-Claude Latombe
June 2002 **Proceedings of the eighteenth annual symposium on Compu
SCG '02**

Publisher: ACM Press

Full text available:  [pdf\(270.11 KB\)](#) Additional Information: [full citation](#), [abst](#)
[citations](#), [index ter](#)

The kinematic chain is a ubiquitous and extensively studied representati well as a useful model for studying the motion of biological macro-mole stand to benefit from algorithms for efficient maintenance and collision c chains. This paper introduces a novel hierarchical representation of a kin allowing for efficient incremental updates and relative position calculati oriented bounding boxes is superimposed on this r ...


Keywords: bounding volume hierarchy, collision detection, modelling K molecular modelling, self collisions

9 Class-based probability estimation using a semantic hierarchy

Stephen Clark, David Weir

June 2002 **Computational Linguistics**, Volume 28 Issue 2


Publisher: MIT Press

Full text available:  [pdf\(422.66 KB\)](#) Additional Information: [full citation](#), [abst](#)
[citations](#), [index ter](#)



This article concerns the estimation of a particular kind of probability, na probability of a noun sense appearing as a particular argument of a predi overcome the accompanying sparse-data problem, the proposal here is to probabilities in terms of senses from a semantic hierarchy and exploit the can be grouped into classes consisting of semantically similar senses. Th focus on the problem of how to determine a suitable ...

10 Vortex: an optimizing compiler for object-oriented languages

◆ Jeffrey Dean, Greg DeFouw, David Grove, Vassily Litvinov, Craig Chamb
October 1996 **ACM SIGPLAN Notices**, **Proceedings of the 11th ACM
conference on Object-oriented programming, systems, la**

applications OOPSLA '96, Volume 31 Issue 10**Publisher:** ACM PressFull text available:  [pdf\(2.45 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)


Previously, techniques such as class hierarchy analysis and profile-guide prediction have been demonstrated to greatly improve the performance of programs written in pure object-oriented languages, but the degree to which these techniques are transferable to applications written in hybrid languages has been unclear. In this question, we have developed the Vortex compiler infrastructure, a language-independent optimizing compiler for object-oriented languages, with ...

11 OBBTree: a hierarchical structure for rapid interference detection S. Gottschalk, M. C. Lin, D. ManochaAugust 1996 **Proceedings of the 23rd annual conference on Computer graphics and interactive techniques SIGGRAPH '96****Publisher:** ACM PressFull text available:  [pdf\(341.04 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)

Keywords: collision detection, contacts, hierarchical data structure, physics modeling, shape approximation, virtual prototyping

12 Usability testing: revisiting informed consent procedures for testing internet-based systems

Oliver K. Burmeister

November 2000 **Selected papers from the second Australian Institute of Computer Ethics CRPIT '00****Publisher:** Australian Computer Society, Inc.Full text available:  [pdf\(753.45 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

This paper explores issues of professional, ethical conduct in usability testing around the concept of 'informed consent'. Previous work on informed consent has been limited to homogeneous geographic locations. With Internet sites being developed worldwide, these procedures need to be revisited for their applicability to heterogeneous terms of culture, business practice, language and legal requirements. Some

valued principles might now be considered discreti ...

13 Structure and transformation of documents: Simple and accurate feature se
hierarchical categorisation



Wahyu Wibowo, Hugh E. Williams

November 2002 **Proceedings of the 2002 ACM symposium on Document Analysis and Recognition**
DocEng '02

Publisher: ACM Press

Full text available: [pdf\(161.40 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Categorisation of digital documents is useful for organisation and retrieval. Categories can be a set of unstructured category labels, some documents are hierarchically structured. This paper investigates automatic hierarchical categorisation, specifically, the role of features in the development of more effective categorisers. It shows that a good hierarchical machine learning-based categoriser can be developed using a large number of features from pre-categorised training data ...

Keywords: categorisation, error reduction, hierarchical categorisation, w

14 Fault classes and error detection capability of specification-based testing



D. Richard Kuhn

October 1999 **ACM Transactions on Software Engineering and Methodology**
Volume 8 Issue 4

Publisher: ACM Press

Full text available: [pdf\(124.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


Some varieties of specification-based testing rely upon methods for generating test cases from predicates in a software specification. These methods derive variations from logic expressions, with the aim of detecting different types of faults. This paper has presented empirical results on the ability of specification-based test cases to detect failures. This article describes a method for computing the coverage of a test set for the test ...

Keywords: testing

15 Design: no job too small

◆ Jean C. Scholtz, Pete Lockhart, Tony Salvador, James Newbery
March 1997 **Proceedings of the SIGCHI conference on Human factors : systems CHI '97**


Publisher: ACM Press

Full text available:  [pdf\(942.85 KB\)](#) Additional Information: [full citation](#), [reference terms](#)

Keywords: design, ergonomics, hand held, mobile computing, testing, user requirements

16 Papers: Operational and performance issues of a CBQ router

◆ Fulvio Risso, Panos Gevros
October 1999 **ACM SIGCOMM Computer Communication Review**, Volume 29 Issue 5
Publisher: ACM Press

Full text available:  [pdf\(1.16 MB\)](#) Additional Information: [full citation](#), [abstracts](#)

The use of scheduling mechanisms like Class Based Queueing (CBQ) is a key role in next generation multiservice IP networks. In this paper we attempt an experimental evaluation of ALTQ/CBQ demonstrating its sensitivity to various parameters and link layer driver design issues. We pay attention to several parameters that affect performance drastically and particularly to "borrowing" for flexible and efficient link sharing. We are also investigating ...

17 On randomization in sequential and distributed algorithms

◆ Rajiv Gupta, Scott A. Smolka, Shaji Bhaskar
March 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 1
Publisher: ACM Press

Full text available:  [pdf\(8.01 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

Probabilistic, or randomized, algorithms are fast becoming as common as deterministic algorithms. This survey presents five techniques that have been used in the design of randomized algorithms. These techniques are illustrated using algorithms—both sequential and distributed—that span a wide range of

including: primality testing (a classical problem in number theory), interactive proofs ...


Keywords: Byzantine agreement, CSP, analysis of algorithms, computational dining philosophers problem, distributed algorithms, graph isomorphism, interactive probabilistic proof systems, leader election, message routing, problem, perfect hashing, primality testing, probabilistic techniques, randomized probabilistic algorithms, randomized quicksort, sequential algorithms, tournaments, universal hashing


18 The interaction of knowledge sources in word sense disambiguation

Mark Stevenson, Yorick Wilks

September 2001 **Computational Linguistics**, Volume 27 Issue 3

Publisher: MIT Press

Full text available:  pdf(347.52

KB) 

Publisher

Site

Additional Information: full citation, abstracts, citations


Word sense disambiguation (WSD) is a computational linguistics task like the tradition of combining different knowledge sources in artificial intelligence. An important step in the exploration of this hypothesis is to determine which knowledge sources are most useful and whether their combination leads to better results. We present a sense tagger which uses several knowledge sources. Tested on 94% on our evaluation corpus. Our system attempts ...

19 Testing and debugging: Some issues in multi-phase software reliability modeling

M. A. Vouk, K. C. Tai

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Collaborative research: software engineering - Volume 1**

Publisher: IBM Press

Full text available:  pdf(810.20 KB) Additional Information: full citation, abstracts, citations

During early software testing phases, testing profiles are often very different from operational profiles. Consequently, assessment of operational software quality during non-operational testing stages is difficult, and is open to interpretation. This paper discusses some issues related to this. Software is assumed to be a large system consisting of many modules.


components that evolve in parallel. The focus is on early identification of components that in operation may be excessively error-prone. ...

20 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Artificial Intelligence on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)




Understanding distributed applications is a tedious and difficult task. Visually, on process-time diagrams are often used to obtain a better understanding of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex, making it difficult for the user with the desired overview of the application. In our experience, repeated occurrences of non-trivial communication ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery
ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)

[Home](#) | [Login](#) | [Logout](#)


Welcome United States Patent and
Trademark Office

Search Results

[BROWSE SEARCH](#) [IEEE GUIDE](#)

Results for "**(((test hierarchy execution flow failure parameter)<in>met
<and> (pyr >= 1990**"
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance
Descending** order.

» Search Options

[View Session
History](#)
[New Search](#)

Modify Search

☐ Check to search only within this results set

» Key

IEEE JNL IEEE
Journal or
Magazine

IEEE JNL IEE Journal
or Magazine

IEEE CNF IEEE
Conference
Proceeding

IEEE CNF IEE
Conference
Proceeding

IEEE STD IEEE
Standard

Display Format: ☒ Citation ☐ Citation &
Abstract

No results were found.

Please edit your search criteria and try again. Refer
assistance revising your search.

Indexed by

Inspect

[Home](#) | [Login](#) | [Logout](#)**IEEE Xplore**
RELEASE 2.1**Welcome United States Patent and
Trademark Office**▣ **Search Results****BROWSE SEARCH IEEE
GUID**

Results for "**((test flow failure parameter)<in>metadata)) <and> (pyr <and> pyr <...>))**"
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance Descending** order.

» **Search Options**[View Session
History](#)[New Search](#)**Modify Search****((test flow failure parameter)<in>metadata)) <and>**☐ Check to search only within this results set» **Key****IEEE
JNL**

IEEE

Journal or
Magazine**IEE
JNL**IEE Journal
or Magazine**IEEE
CNF**

IEEE

Conference
Proceeding**IEE
CNF**

IEE

Conference
Proceeding**IEEE
STD**

IEEE

Standard

**Display
Format:** ☒ Citation ☐ Citation &
Abstract**No results were found.**Please edit your search criteria and try again. Refer
assistance revising your search.

Indexed by

 **Inspec**



test hierarchy execution flow failure

1990

- 2

Scholar [All articles](#) [Recent articles](#) Results 1 - 10 of about 6,750 for **test hie**

All Results [System and method for automated testing and monitoring of software applications - group of 3 »](#)
[O Balci](#) UHH Wild, MI Jabri - US Patent 5,671,351, 1997 - Google
[J Horgan](#) Patents
[D Le](#) ... 6 US Patent Sep. 23, 1997 Sheet 4 of 15 5,671,351 Control
[P Oman](#) Parameters **Test Execution**
[R Taylor](#) Stop on **Failure** Iteration Counl Delay Time (sec) Apply K
 7 : 1 Controlled ...
[Cited by 23](#) - [Related Articles](#) - [Web Search](#)

[Using reliability models during testing with non-operational profiles - group of 4 »](#)
 MA Vouk - Computer Science Department, North Carolina State University, 1992 - renoir.csc.ncsu.edu
 ... model each **test** case exercises a **hierarchy** of functions ...
 be the CPU **execution** time,
 calendar **execution** time ... **test** cases executed, the number of
test cases executed ...
[Cited by 17](#) - [Related Articles](#) - [View as HTML](#) - [Web Search](#)

[Simplifying TPS development and **execution** using a PC, Web-based environment - group of 4 »](#)
 D Rolince, T Inc, MA Boston - AUTOTESTCON'98. IEEE Systems Readiness Technology Conference ..., 1998 - ieexplore.ieee.org
 ... all the other nodes at that level of **hierarchy** and lower ...
 the course of **test** program
 integration and debug, **test** program **execution flow**
 frequently needs ...
[Web Search](#) - [BL Direct](#)

Some Critical Remarks on a **Hierarchy** of Fault-Detecting Abilities of Test Methods - group of 9 »

RA DeMillo, AP Mathur, WE Wong - IEEE Transactions on Software Engineering, 1995 - doi.ieeecomputersociety.org

... casual reader to conclude that a **hierarchy** of **test** ... is a special statement that upon **execution** causes the ... mutation and data **flow** based on **test** adequacy criteria ...

Related Articles - Web Search - BL Direct

ABET-an architecture for integrated diagnostics?

PM McCown, ASA Co, NJ Teterboro - AUTOTESTCON'92. IEEE Systems Readiness Technology Conference ..., 1992 - ieeexplore.ieee.org

... ATLAS DATA TyPES FIGURE 2 **HIERARCHY** OF DATA ... the control of task **execution** prioritization, which is ... sources (situation assessment) for **test** strategy development ...

Web Search - BL Direct

Static and Dynamic Analysis of Programs that Contain Arbitrary Interprocedural Control **Flow** - group of 3 »

S Sinha - 2002 - cc.gatech.edu

... tasks, including **test**-coverage analysis, **test**-data generation, **execution** profiling, debugging, and ... Finally, we develop a **hierarchy** of **test**-selection and **test** ...

Cached - Web Search - Library Search

Achieving software quality with testing coverage measures - group of 5 »

JR Horgan, S London, MR Lyu - Computer, 1994 - ieeexplore.ieee.org

... In Figure 1, an example of a du-path would be **execution** of a ... these control or dataflow constructs are covered if they execute during the **test**.

Hierarchy. ...

Cited by 62 - Related Articles - Web Search - BL Direct

Metrics for assessing a software system's maintainability

P Oman, J Hagemester - Software Maintenance, 1992.
Proceedings., Conference on, 1992 - ieeexplore.ieee.org
... test coverage, Complexity of test procedures,
Installation ... intramodular control flow
and execution of a ... subtree of the maintainability
hierarchy: a. Complexity ...

[Cited by 42](#) - [Related Articles](#) - [Web Search](#)

Structural testing of concurrent programs - group of 8 »

RN Taylor, DL Levine, CD Kelly - Software Engineering,
IEEE Transactions on, 1992 - ieeexplore.ieee.org
... concurrency criteria are with respect to the execution state
space ... 1. Concurrency
coverage criteria subsumption hierarchy. ... the program
over a set of test data. ...

[Cited by 92](#) - [Related Articles](#) - [Web Search](#)

Verification, validation, and accreditation

O Balci - Proceedings of the 30th conference on Winter
simulation, 1998 - portal.acm.org
... Execution Tracing ... The multifaceted methodology
employs a hierarchy of hundreds of
indicators, analytic hierarchy process (AHP) for ... Real-
Time Input Testing ...

[Cited by 36](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Go o o o o o o o o o o g l e ►

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

©2007 Google



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service\)](#)
[Search:](#) ☒ The ACM Digital Library ☐ The

[+test +execution +flow +expected](#)
[Feedback](#) [Report a problem](#)

Published since January 1990 and Published before
November 2003

Terms used **test** **execution** **flow** **expected**

Sort results
by

☒ [Save results to a Binder](#)
[Try an Advanced](#)
☒ [Search Tips](#)
[Try this search](#)

Display
results

☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Results

1 [An analytical comparison of the fault-detecting ability of data flow testing](#)

Phyllis G. Frankl, Elaine J. Weyuker

May 1993 **Proceedings of the 15th international conference on Software Engineering (ICSE '93)**

Publisher: IEEE Computer Society Press

Full text available: ☒ [pdf\(1.05 MB\)](#)

Additional Information: [full citation](#), [references](#)

2 [Automated test oracles for GUIs](#)

Atif M. Memon, Martha E. Pollack, Mary Lou Soffa

November 2000 **ACM SIGSOFT Software Engineering Notes**, **Proceedings of the ACM SIGSOFT international symposium on Foundations of software engineering: twenty-first century applications SIGSOFT '00**
Volume 25 Issue 6

Publisher: ACM Press

Full text available: ☒ [pdf\(1.29 MB\)](#)

Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

Graphical User Interfaces (GUIs) are critical components of today's software. GUIs have different characteristics than traditional software, conventional testing techniques

not apply to GUI software. In previous work, we presented an approach cases, which take the form of sequences of actions. In this paper we develop a technique to determine if a GUI behaves as expected for a given test case. A formal model of a GUI, expressed as se ...


Keywords: GUI test oracles, GUI testing, automated oracles

3 A comparative study of coarse- and fine-grained safe regression test-selection

John Bible, Gregg Rothermel, David S. Rosenblum

April 2001 **ACM Transactions on Software Engineering and Methodology**
Volume 10 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(204.13 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

Regression test-selection techniques reduce the cost of regression testing by selecting a subset of an existing test suite to use in retesting a modified program. Over the past decades, numerous regression test-selection techniques have been described. Initial empirical studies of some of these techniques have suggested that they can benefit testers, but so far, few studies have empirically compared different techniques. In this paper, we present ...


Keywords: regression test selection, regression testing

4 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Research on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizing process-time diagrams are often used to obtain a better understanding of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience,


repeated occurrences of non-trivial commun ...

5 An empirical investigation of program spectra

◆ Mary Jean Harrold, Gregg Roethermel, Rui Wu, Liu Yi

July 1998 **ACM SIGPLAN Notices , Proceedings of the 1998 ACM SIG workshop on Program analysis for software tools and engine**
Volume 33 Issue 7

Publisher: ACM Press

Full text available:  [pdf\(965.56 KB\)](#) Additional Information: [full citation](#), [abst](#)
[citions](#), [index ter](#)

A variety of expensive software maintenance and testing tasks require a behaviors of program versions. Program spectra have recently been prop for use in performing such comparisons. To assess the potential usefulne context, we conducted an experiment that examined the relationship betw spectra and program behavior, and empirically compared several types o reports the results of that experiment.


6 Analysis and testing of Web applications

Filippo Ricca, Paolo Tonella

July 2001 **Proceedings of the 23rd International Conference on Softwa**
ICSE '01

Publisher: IEEE Computer Society

Full text available:  [pdf\(167.58](#)

[KB\)](#) 

[Publisher](#)

[Site](#)

Additional Information: [full citation](#), [abst](#)
[citions](#), [index ter](#)

The economic relevance of Web applications increases the importance o improving their quality. Moreover, the new available technologies for th allow the insertion of sophisticated functions, but often leave the develop their organization and evolution. As a consequence, a high demand is em methodologies and tools for quality assurance of Web based systems.

In this paper, a UML model of Web applications is proposed for their ...


Keywords: UML modeling, code analysis, reverse engineering, testing,

7 Experimental evaluation of a fuzzy-set based measure of software correctness mutation

Farokh B. Bastani, Giuseppe DiMarco, Alberto Pasquini

May 1993 **Proceedings of the 15th international conference on Software Engineering ICSE '93**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(1.00 MB\)](#)

Additional Information: [full citation](#), [references](#)


8 The privatizing DOALL test: a run-time technique for DOALL loop identification and privatization



Lawrence Rauchwerger, David Padua

July 1994 **Proceedings of the 8th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.27 MB\)](#)

Additional Information: [full citation](#), [abstracts](#), [index terms](#)


Current parallelizing compilers cannot identify a significant fraction of for loops because they have complex or statically insufficiently defined access patterns. In this paper, we describe a technique for identifying for loops that can be executed in parallel. Because, for a given reason, we have developed the Privatizing DOALL test—a technique for identifying for loops that can be executed in parallel at run-time, and dynamically privatizing scalars and arrays used in the loops. The technique is simple, and can be applied to any loop, regardless of the loop's control flow. The technique ...

9 On testing of classes in object-oriented programs

Dechang Gu, Yin Zhong, Sarwar Ali

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Research in Software Engineering CASCON '94**

Publisher: IBM Press

Full text available:  [pdf\(44.44 KB\)](#)

Additional Information: [full citation](#), [abstracts](#), [index terms](#)

Object-oriented technology has been widely studied and applied. Substantial research has been carried out in object-oriented analysis, design and programming languages. However, relatively little attention has been paid to testing of object-oriented programs. In this paper, we review several techniques proposed in the literature for testing object-oriented programs and investigate the impact of object-oriented approach on the design of testing techniques.


particular, we focus on the test case ...

10 Efficient instrumentation for code coverage testing

Mustafa M. Tikir, Jeffrey K. Hollingsworth

July 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of SIGSOFT international symposium on Software testing and '02, Volume 27 Issue 4**

Publisher: ACM Press

Full text available:  [pdf\(524.54 KB\)](#) Additional Information: [full citation](#), [abstracts](#)

Evaluation of Code Coverage is the problem of identifying the parts of a program that do not execute in one or more runs of a program. The traditional approach for code coverage tools is to use static code instrumentation. In this paper we present a new technique to dynamically insert and remove instrumentation code to reduce the runtime overhead of code coverage. We also explore the use of dominator tree information to reduce the number of instrumentation points needed. Our experiments show that ...


Keywords: code coverage, dominator tree, dynamic code deletion, dynamic on-demand instrumentation, testing

11 Augmenting data flow criteria for class testing

Pei Hsia, Xiaolin Li, David C. Kung

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Research on Collaborative research CASCON '97**

Publisher: IBM Press

Full text available:  [pdf\(219.68 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

A class is widely considered the basic unit of testing in object-oriented systems. Although there has been much recent research on class testing, little attention has been paid to data flow based class testing criteria. In this paper, we extend the traditional data flow based testing techniques and propose three new code-based class testing criteria. These criteria overcome the problems associated with existing data flow techniques. We show that each of the new criteria is stricter than its ...

12


Special session on on-chip multi-processing: Design experience of a chip n

◆ merlot and expectation to functional verification

Satoshi Matsushita

October 2002 **Proceedings of the 15th international symposium on Syst
'02**

Publisher: ACM Press

Full text available:  [pdf\(797.44 KB\)](#) Additional Information: [full citation](#), [abst](#)
[citations](#), [index ter](#)

We have fabricated a Chip Multiprocessor prototype code-named Merlot speculative multithreading architecture. On Merlot, multiple threads pro window beyond ordinal instruction level parallel (ILP) processors like su With the architecture, we estimate 3.0 times speedup against single proci (PE) on speech recognition code and IDCT code with four PEs. Merlot i devices, PCI interface, and SDRAM interfaces. We have en ...



Keywords: CMP, chip multiprocessor, deign experience, functional veri multithreading

13 Software acceleration using programmable logic: is it worth the effort?

Martyn Edwards

March 1997 **Proceedings of the 5th International Workshop on Hardw:
Design CODES '97**

Publisher: IEEE Computer Society



Full text available:  [pdf\(706.30 KB\)](#) 

[Publisher](#)
[Site](#)


Additional Information: [full citation](#), [abst](#)

A commonly accepted technique in hardware/software co-design is to in system functions as possible in software and to move performance critica special-purpose external hardware in order to either satisfy timing constr overall execution time of a program - this is known as "software accelera investigates the limits to the performance enhancements obtainable using acceleration techniques. A practical target architecture, b ...

Keywords: software acceleration, performance evaluation, hardware arc

14 Verification of communication protocols using data flow analysis **Gleb N. Naumovich, Lori A. Clarke, Leon J. Osterweil****October 1996 ACM SIGSOFT Software Engineering Notes , Proceedings
SIGSOFT symposium on Foundations of software engineering
'96, Volume 21 Issue 6****Publisher:** ACM PressFull text available:  [pdf\(1.39 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

In this paper we demonstrate the effectiveness of data flow analysis for verifying the requirements of communication protocols. Data flow analysis is a static analysis that is increasing confidence in the correctness of software systems by automatically checking that a given software artifact (e.g., design or code) must behave consistently with its requirements. In this case study, we apply the FLAVERS data flow analysis to the pseudocode designs of the three way handshake connection establishment protocol.

15 The LRPD test: speculative run-time parallelization of loops with privatization **Lawrence Rauchwerger, David Padua****June 1995 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN
on Programming language design and implementation PLDI
Issue 6****Publisher:** ACM PressFull text available:  [pdf\(1.74 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

Current parallelizing compilers cannot identify a significant fraction of parallelizable loops because they have complex or statically insufficiently defined access patterns. Since parallelizable loops arise frequently in practice, we advocate a novel framework for their identification: speculatively execute the loop as a doall, and apply a fully static dependence test to determine if it had any cross-iteration dependences; if the loop is re-executed serially. Since ...

16 The enable construct for exception handling in Fortran 90 **CORPORATE IFIP Working Group 2.5****October 1993 ACM SIGNUM Newsletter, Volume 28 Issue 4****Publisher:** ACM PressFull text available:  [pdf\(503.87 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

KB)terms


It is apparent that a good mechanism for handling computing exceptions construction of robust and maintainable Fortran 90 code. Ideally such co rapidly when all is well and not fail in situations such as when data is ne limits or there is insufficient memory for an automatic array. A language that permits transfer of control when necessary from straightforward fast code that uses an alternative algorithm.

17 Session 3B: Software testing: Local exhaustive testing: a software reliabili

◆ Thomas Wood, Keith Miller, Robert E. Noonan

April 1992 **Proceedings of the 30th annual Southeast regional conferen**

Publisher: ACM Press

Full text available:  [pdf\(432.04 KB\)](#) Additional Information: [full citation](#), [abst](#)
[citations](#)


We introduce local exhaustive testing as a simple strategy for creating te uncover faults (a deficiency in the code that is responsible for incorrect t higher probability than tests chosen randomly. To use local exhaustive te certain inputs points as "critical," and then test all inputs close to that poi this strategy will be particularly effective in applications that include an geometric or other regular organization. ...

18 Code generation and analysis for the functional verification of micro proce

◆ Anoosh Hosseini, Dimitrios Mavroidis, Pavlos Konas

June 1996 **Proceedings of the 33rd annual conference on Design autom.**

Publisher: ACM Press


Full text available:  [pdf\(87.31 KB\)](#) Additional Information: [full citation](#), [refe](#)
[index terms](#)

19 Grading student programs - a software testing approach

Edward L. Jones


November 2000 **Journal of Computing Sciences in Colleges , Proceedin**
fourteenth annual consortium on Small Colleges South
CCSC '00 , Proceedings of the second annual CCSC or
Small Colleges Northwestern conference, Volume 16 Is

Publisher: Consortium for Computing Sciences in Colleges

Full text available:  [pdf\(36.40 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)


This paper describes an experience of automating the grading of student programs. The framework provides guidance for developing the assignment specific program. Automation saves time and improves grading consistency and accuracy for students. After an adjustment period, student programs improved. Although it invests more time writing a testable assignment specification and developing the program, these costs are expected to be amortized over multiple years.

20 [A comprehensive approach to parallel data flow analysis](#)

 Yong-Fong Lee, Barbara G. Ryder

August 1992 **Proceedings of the 6th international conference on Supercomputing**

Publisher: ACM Press

Full text available:  [pdf\(1.27 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [index terms](#)

We present a comprehensive approach to performing data flow analysis to identify three types of parallelism inherent in the data flow solution process: problem parallelism, separate-unit parallelism and algorithmic parallelism. We develop a unified framework to exploit them. Our investigations of typical Fortran programs show an abundance of the last two types of parallelism. In particular, we illustrate the use of algorithmic parallelism in the context of data flow analysis.




Keywords: data flow analysis, parallel algorithms, parallel data flow analysis

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery
ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)



test hierarchy execution flow failure

1990

- 2003



Scholar All articles Recent articles Results 1 - 10 of about 6,750 for test hierarchy execu

All Results

O Balci

J Horgan

D Le

P Oman

R Taylor

System and method for automated testing and monitoring of software applications - group of 3 »

UHH Wild, MI Jabri - US Patent 5,671,351, 1997 - Google Patents

... 6 US Patent Sep. 23, 1997 Sheet 4 of 15 5,671,351 Control Parameters

Test Execution

Stop on **Failure** Iteration Counl Delay Time (sec) Apply K 7 : 1

Controlled ...

Cited by 23 - Related Articles - Web Search



Using reliability models during testing with non-operational profiles - group of 4 »

MA Vouk - Computer Science Department, North Carolina State University, 1992 - renoir.csc.ncsu.edu

... model each **test** case exercises a **hierarchy** of functions ... be the CPU **execution** time,

calendar **execution** time ... **test** cases executed, the number of **test** cases executed ...

Cited by 17 - Related Articles - View as HTML - Web Search

Simplifying TPS development and **execution** using a PC, Web-based environment - group of 4 »

D Rolince, T Inc, MA Boston - AUTOTESTCON'98. IEEE Systems Readiness Technology Conference ..., 1998 - ieexplore.ieee.org

... all the other nodes at that level of **hierarchy** and lower ... the course of **test** program

integration and debug, **test** program **execution** flow frequently needs ...

Web Search - BL Direct

Some Critical Remarks on a **Hierarchy** of Fault-Detecting Abilities of **Test** Methods - group of 9 »

RA DeMillo, AP Mathur, WE Wong - IEEE Transactions on Software Engineering, 1995 - doi.ieeecomputersociety.org

... casual reader to conclude that a **hierarchy** of **test** ... is a special statement that upon **execution** causes the ... mutation and data **flow** based on **test** adequacy criteria ...

[Related Articles](#) - [Web Search](#) - [BL Direct](#)

ABET-an architecture for integrated diagnostics?

PM McCown, ASA Co, NJ Teterboro - AUTOTESTCON'92. IEEE Systems Readiness Technology Conference ..., 1992 - [ieeexplore.ieee.org](#)
... ATLAS DATA TyPES FIGURE 2 **HIERARCHY** OF DATA ... the control of task **execution** prioritization, which is ... sources (situation assessment) for **test** strategy development ...
[Web Search](#) - [BL Direct](#)

Static and Dynamic Analysis of Programs that Contain Arbitrary Interprocedural Control Flow - [group of 3 »](#)

S Sinha - 2002 - [cc.gatech.edu](#)

... tasks, including **test**-coverage analysis, **test**-data generation, **execution** profiling, debugging, and ... Finally, we develop a **hierarchy** of **test**-selection and **test** ...

[Cached](#) - [Web Search](#) - [Library Search](#)

Achieving software quality with testing coverage measures - [group of 5 »](#)

JR Horgan, S London, MR Lyu - Computer, 1994 - [ieeexplore.ieee.org](#)
... In Figure 1, an example of a du-path would be **execution** of a ... these control or dataflow
con- structs are covered if they execute dur- ing the **test**. **Hierarchy**. ...
[Cited by 62](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Metrics for assessing a software system's maintainability

P Oman, J Hagemeister - Software Maintenance, 1992. Proceedings., Conference on, 1992 - [ieeexplore.ieee.org](#)
... **test** coverage, Complexity of **test** procedures, Installation ...
intramodular control **flow**
and **execution** of a ... subtree of the maintainability **hierarchy**: a. Complexity ...

[Cited by 42](#) - [Related Articles](#) - [Web Search](#)

